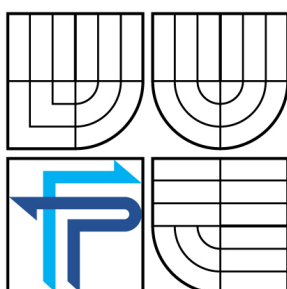


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ**  
**ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

# NÁVRH DATOVÉ APLIKACE PRO FIRMU PEKÁRNA TANVALD

DATA APPLICATION CONCEPT FOR COMPANY PEKÁRNA TANVALD

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**PETR SCHNEIDER**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Ing. JIŘÍ KŘÍŽ, Ph.D.**

BRNO 2008

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

**Schneider Petr**

---

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

**Návrh datové aplikace pro firmu Pekárna Tanvald**

v anglickém jazyce:

**Data application concept for company Pekárna Tanvald**

Pokyny pro vypracování:

Úvod  
Vymezení problému a cíle práce  
Teoretická východiska práce  
Analýza problému a současné situace  
Vlastní návrhy řešení, přínos návrhů řešení  
Závěr  
Seznam použité literatury  
Přílohy

Seznam odborné literatury:

PÍSEK, Slavoj. Access 2000. 1999. ISBN: 80-7169-880-6

RIESSLER, Petr. Databáze a programování. 2000. ISBN: 80-214-1778-1

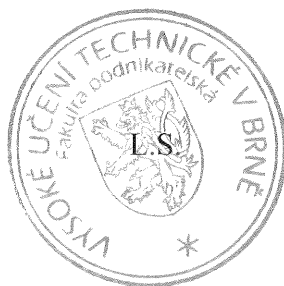
MERUNKA, Vojtěch. Objektový přístup v databázových systémech. 2002. ISBN: 80-213-0882-6

VAUGHN William R.. Visual Basic pro SQL server : průvodce tvorbou databázových aplikací. 1998. ISBN: 80-7226-085-5

RIORDAN Rebecca M.. Vytváříme relační databázové aplikace. 2000. ISBN: 80-7226-360-9

Vedoucí bakalářské práce: Ing. Jiří Kříž, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2007/08.



---

Ing. Jiří Kříž, Ph.D.  
Ředitel ústavu

---

doc. Ing. Miloš Koch, CSc.  
Děkan fakulty

V Brně, dne 15.2.2008

## **Abstrakt**

Bakalářská práce se zabývá tvorbou databázového systému pro firmu Pekárna Tanvald podnikající v potravinářském odvětví, jenž by měl pomoci při náhradě papírových spisů a zlepšit majiteli přehlednost.

## **Abstract**

This bachelor's thesis deal with concept data application for company Pekárna Tanvald which makes business in food-processing industry, This concept should help in the replacement of paper writting documents and make better lucidity for owner.

## **Klíčová slova**

PHP, MySQL, entita, atribut, identifikátor, vazba.

## **Key words**

PHP, MySQL, entity, attribute, identifier, structure.

## **Bibliografická citace práce**

SCHNEIDER, P. *Návrh datové aplikace pro firmu Pekárna Tanvald*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2008. 55 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

## **Čestné prohlášení**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 24. května 2008

-----

Podpis

## **Poděkování**

Chtěl bych poděkovat svému vedoucímu panu Ing. Jiřímu Křížovi, Ph.D., za jeho ochotu a pomoc při vypracování této bakalářské práce, a také oponentovi Mgr. Pavlu Šafářovi, který obětoval svůj čas na prostudování a zhodnocení mé práce.

# Obsah

<b>1. Úvod .....</b>	<b>9</b>
<b>2. Vymezení problému a cíle práce.....</b>	<b>10</b>
<b>3. Teoretická východiska práce .....</b>	<b>11</b>
3.1. Teoretické poznatky .....	11
3.2. PHP .....	12
3.3. Historie .....	12
3.3.1. PHP/FI .....	12
3.3.2. PHP 3 .....	13
3.3.3. PHP 4 .....	13
3.3.4. PHP 5 .....	14
3.4. Jazyk SQL .....	14
3.5. Databázový systém MySQL .....	15
3.6. Apache.....	16
<b>4. Analýza problému a současné situace.....</b>	<b>17</b>
4.1. Stručná historie firmy, výhled do budoucnosti .....	17
4.2. Organizační struktura .....	18
4.3. Organizační procesy .....	19
4.4. HW profil firmy .....	22
4.5. SW profil firmy .....	23
4.6. Shrnutí analýzy.....	25
<b>5. Vlastní návrhy řešení.....</b>	<b>26</b>
5.1. Databázové požadavky na systém.....	26
5.2. Základní požadavky na aplikaci.....	26
5.3. Základní popis entit.....	27
5.4. Normalizace .....	29
5.5. Entitě relační diagram .....	31
5.6. Definice atributů entit .....	33
5.7. Vytvoření tabulek.....	37
5.8. Návrh systému pro docházku .....	43
5.9. Řešení webové aplikace .....	44



5.9.1.	Formulář přihlášení.....	44
5.9.2.	Administrační část .....	47
5.9.3.	Zaměstnanec .....	47
5.9.4.	Automobil .....	48
5.9.5.	Kniha jízd.....	49
5.9.6.	Přehled .....	50
<b>6.</b>	<b>Přínos návrhů řešení.....</b>	<b>51</b>
<b>7.</b>	<b>Závěr .....</b>	<b>52</b>
<b>8.</b>	<b>Seznam použité literatury .....</b>	<b>53</b>
8.1.	Literatura .....	53
8.2.	Internetové zdroje.....	54
<b>9.</b>	<b>Seznam obrázků.....</b>	<b>55</b>

# 1. Úvod

Uchovávání informací mělo v průběhu minulých tisíciletí různé podoby. Vzpomeňme na nástěnné malby v jeskyních našich prapředků, hieroglyfy na stěnách hrobek faraónů či hliněné desky s klínovým písmem. Od té doby uplynulo mnoho času a uchovávání dat doznalo jistých změn. Přelomem se stal vynález papíru jakožto media pro zápis událostí, které se k tomuto účelu používá dodnes, i když již v jiné než v původní podobě. V současné době, kdy informační technologie kráčí kupředu mílovými kroky, nahrazuje papír a pero v mnoha případech počítač a elektronické dokumenty.

Každý z nás si zcela určitě vybaví, jakým způsobem byla uchovávána velká množství dat donedávna. Velké plechové boxy s šuplíky plnými papírových složek seřazených v abecedním pořadí plnily podnikové archivy, kanceláře či ordinace našich lékařů.

S příchodem moderní technologie je možné obsah takového archivu uložit na paměťové médium, ať už optické (CD, DVD), magnetické (HDD) či na jejich kombinaci magnetooptické disky. A díky tomu na místo plné místnosti papíru, můžeme všechny informace nosit v brašně svého notebooku a mít vše stále s sebou. Velké i malé kartotéky pomalu přebírají elektronickou podobu ať ve formě malých databází či velkých datových skladů.

## **2. Vymezení problému a cíle práce**

Cílem této bakalářské práce je vytvořit návrh datové aplikace s možností rozšíření podle potřeb do budoucnosti, na jehož základě bude možné upustit od ukládání dat v podobě papírových spisů a nahradit tak stávající kartotéku aplikací, jenž umožní efektivnější vyhledávání a poskytne větší přehlednost.

### 3. Teoretická východiska práce

#### 3.1. Teoretické poznatky

**Entita** - je rozlišitelný a identifikovatelný objekt reálného světa, který je předmětem našeho zájmu a o kterém má smysl podávat informace.[18]

Vymezení entity je dosti volné (totéž platí pro pojmy, vztah a atribut), neexistuje jednoznačné pravidlo, zda zvolit data jako entitu či jako vztah. Často závisí na úhlu pohledu analytika.[18]

**Identifikace entity** - každá entitní množina musí mít uveden identifikátor, tj. minimální množinu atributů, které zajišťují jednoznačnou identifikaci entit v dané množině. Pojmu IDENTIFIKÁTOR ENTITY odpovídá na technologické a implementační úrovni pojem PRIMÁRNÍ KLÍČ.[15]

**Atributy** – funkce přiřazující entitám či vztahům hodnotu, určující některou podstatnou vlastnost entity nebo vztahu atribut je VLASTNOST ENTITY. Je to datový prvek blíže charakterizující entitu, vztah.[19]

**Klíče:** *identifikátor/primární klíč* je minimální množina atributů zajišťující jednoznačnou identifikaci výskytů entity, *cizí klíč* je atribut nebo množina atributů, které jsou v jiné entitě primárním klíčem nebo jeho částí. V datovém modelu slouží pro vyjádření vztahů mezi entitami.

#### **Vztah mezi entitami**

Jednotlivé entity vstupují do konkrétních vzájemných vztahů:

- vztah vyjadřuje reálnou vazbu mezi dvěma a více entitami
- vztah vyjadřuje informaci, kterou nelze odvodit z atributů jednotlivých entit

- vztah má jméno, vyjadřující podstatu vztahu z hlediska obou partnerských entit i z hlediska vztahu jako takového [19]

### **3.2.     *PHP***

Zkratka PHP byla dříve používána místo výrazu Personal Home Page a jazyk skrývající se za ní, byl používán převážně k realizaci formulářů užívaných na webu. PHP je skriptovací programovací jazyk pro tvorbu dynamického webu, který umožňuje snadné programování na straně serveru (server-side programming). Skript napsaný v PHP vygeneruje na straně serveru stránku a ta je posléze odeslána zpět klientu. Syntaxe jazyka je podobná jazyku C. PHP umožňuje procedurální i objektově orientovaný přístup, takže nemusíme předem definovat typ proměnných a proměnné mohou kdykoliv měnit svůj typ.[16]

Jazyk dále podporuje širokou řadu technologií, formátů a protokolů (HTTP, SMTP, FTP, IMAP, POP3, LDAP...), je to otevřený projekt (ale s dobrou podporou a dokumentací). Dle mého názoru je to v dnešní době jedna z nejpoužívanějších technologií pro tvorbu menších a středních internetových obchodů, půjčoven, portálů, podnikových informačních systémů, redakčních systémů a samozřejmě také drobností typu ankety, počítadla a mnoho jiných. Mnoho z těchto projektů spadá pod svobodnou licenci a kód se dá použít ve vlastních projektech. PHP dobře spolupracuje s webovým serverem Apache a snadno komunikuje s databázemi různých typů (MySQL, PostgreSQL a dalšími).[16]

### **3.3.     *Historie***

#### **3.3.1. PHP/FI**

Začátky PHP se objevují okolo roku 1994. Tehdy se Rasmus Lerdorf rozhodl, že by rád evidoval přístup na své webové stránky. Napsal za tímto účelem jednoduchý systém

v Perlu, který se zalíbil ostatním uživatelům serveru. Systém se začal rozšiřovat, byl doplněn o dokumentaci a uvolněn pod výše již zmíněným názvem Personal Home Page. Rasmus Lerdorf dále vytvořil nástroj, který umožňoval začlenění SQL dotazů do stránek, tvorbu formulářů a práci s daty, která z nich byla získávána. Tento nástroj byl pojmenován Form Interpreter (FI) a byla přidána také podpora MySQL. Okolo roku 1997 přepsali Zeev Suraski a Andi Gutmans hlavní parser PHP/FI 2.0 do jazyka C. Výsledek jejich práce používalo okolo tisíce uživatelů z přibližně 50000 domén oznamujících nainstalované PHP/FI – v té době zhruba 1% všech domén na internetu. Tímto byly položeny základy PHP3.[13]

### **3.3.2. PHP 3**

PHP3 byla první verze, která se blížila PHP, jak ho známe dnes. U této verze se objevilo mnoho rozšíření. Přibyla pevná infrastruktura pro množství různých databází, protokolů a API. To přilákalo také mnoho vývojářů, kteří přidávali vlastní vytvořené rozšiřující moduly. Objevila se také první podpora objektově orientované syntaxe a mnohem silnější a více konzistentní syntaxe samotného jazyka. Pravděpodobně právě proto bylo PHP3.0, v době jeho vrcholu, nainstalováno na přibližně 10% všech serverů Internetu. PHP3.0 bylo oficiálně uvolněno v červnu roku 1998 poté, co bylo veřejně testováno po dobu 9 měsíců.[13]

### **3.3.3. PHP 4**

Krátce po oficiálním uvolnění PHP 3.0 začala autorská dvojice předchozího projektu pracovat na přepsání jádra PHP. Za cíl si zvolili zvýšení výkonu pro složitější aplikace a zlepšení modularity kódové báze PHP. Nový engine, který byl nazván Zend Engine (označení Zend vzniklo složením slabik z křestních jmen autorů, Zeev a Andi), úspěšně splnil očekávání a stanovené cíle a byl uveden v polovině roku 1999. PHP 4.0 založené na tomto enginu bylo oficiálně uvolněno v květnu roku 2000, tedy necelé dva roky po vzniku svého předchůdce. Kromě podstatně zvýšeného výkonu této verze přibýlo i mnoho klíčových prvků. Byla přidána podpora pro mnoho WWW serverů, sessions,

buffering výstupů, bezpečnější zpracování vstupů uživatele a mnoho jazykových konstruktorů. PHP4 používali stovky tisíc vývojářů a nainstalované PHP hlásilo přes 20% domén internetu.[13]

### **3.3.4. PHP 5**

Vývoj PHP 5 začal již v roce 2002. Základem této verze je zcela přepracovaný Zend Engine 2 přinášející vyšší výkon aplikací a umožňující zakomponování pokročilejších programových struktur. V této verzi se výrazně zlepšil přístup k objektově orientovanému programování, který je podobný přístupu v jazyce Java. PHP lze nyní používat jako skutečně objektově orientovaný jazyk.[13]

## **3.4. Jazyk SQL**

Jazyk SQL (Structured Query Language, strukturovaný dotazovací jazyk) je neprocedurální programovací jazyk pro kompletní práci s relačními databázemi, který lze užít snad téměř se všemi databázovými systémy. Samotný jazyk můžeme rozdělit na dvě základní podmnožiny: jazyk DDL (Data Definition Language - jazyk pro popis datové struktury) a DML (Data Manipulation Language - jazyk pro manipulaci s daty). Příkazy z podmnožiny DDL umožňují definici datových struktur a tvorbu objektů, jako jsou tabulky, pohledy, indexy a podobně. Umožňují rovněž měnit jejich strukturu nebo je odstraňovat. Příkazy z podmnožiny DML umožňují manipulaci s daty, to znamená výběr a vkládání dat, jejich aktualizaci a mazání.[14]

První kapitola historie jazyka SQL byla napsána již v roce 1970, v té době ještě pod označením Sequel. Úplně poprvé byl použit pro interní firemní systém (System R), vyvinutý v kalifornské laboratoři IBM. V roce 1979 byl uveden na trh relační databázový systém Oracle firmy Relational Software, Inc. (dnešní Oracle Corporation). Další vývoj jazyka byl poměrně divoký. Na trhu se v první polovině 80. let objevily asi čtyři desítky komerčních SQL systémů.[14]

První významnější produkty, kde byl jazyk SQL zabudován, byly od firmy IBM: DB2, SQL/DS (oba vycházejí ze Systému R) a QMF. S dalším nástupem PC přibýlo více firem, které se začaly o SQL zajímat, například firma Ashton-Tate se svým dBASE IV (implementace SQL nebyla úplná). Dále to byly INFORMIX-SQL, INGRES, SQLBase a XDB II. Proto se přistoupilo k prvnímu normalizačnímu kroku a v roce 1986 byla přijata standardizační skupinou ANSI (o rok později ISO) varianta společnosti IBM pod názvem ANSI SQL (nebo také SQL86).[19]

Další přijatý standard je z roku 1992 (ANSI) a je označován jako SQL92. Zatím nejnovějším standardem je SQL3 (SQL99). Proces standardizace je sice prospěšný, ale nese s sebou i jeden závažný problém – a to zpětnou kompatibilitu.[14]

### **3.5. Databázový systém MySQL**

Jak již bylo uvedeno výše, pro svou práci jsem si vybral databázové prostředí MySQL. Vybral jsem si jej pro jeho snadnou implementaci a také proto, že se jedná o volně šiřitelný software. Je to relační multiplatformní databázový systém typu RDBMS (Relational database management system), vytvořený švédskou firmou MySQLAB. Každá databáze v MySQL je tvořena z jedné nebo více tabulek, které mají řádky a sloupce. Jeden řádek vždy reprezentuje jeden záznam v tabulce. Sloupce jsou pojmenované a uvozují datový typ jednotlivých polí záznamů. MySQL je využitelné nejen v PHP, ale také v C, C++, Java, Perl, Python, Tcl, Visual Basic, NET. [19]

V porovnání s jinými systémy bylo původně MySQL lehce zjednodušené. Obsahovalo pouze jednoduché způsoby zálohování a až donedávna nepodporovalo pohledy, trigger a uložené procedury. Tyto vlastnosti jsou přidávány teprve v posledních letech, kdy by se bez nich vývojáři dynamických webových aplikací obešli jen velmi obtížně. Aktuální verze MySQL 5.0 již podporuje funkce cizích klíčů, transakce, poddotazy, uložené procedury, trigger i pohledy. Ovšem vzhledem k tomu, že zatížení systému nebude enormní, a k soupeření docházet také nebude, nebylo nutné trigger použít.[14]



### **3.6.     *Apache***

Apache HTTP Server je softwarový webový server s otevřeným kódem pro Linux, BSD, Microsoft Windows a další platformy. Vývoj Apache začal v roce 1993 v NCSA (National Center for Supercomputing Applications) na Illinoiské univerzitě. Původní jméno projektu bylo NCSA HTTPd. V dalším roce však vývojářský tým opustil hlavní programátor Rob McCool, díky čemuž došlo ke zpomalení vývoje a v roce 1998 k úplnému zastavení. NCSA HTTPd však mezitím už používali správci webových serverů a dodávali k němu vlastní úpravy. Hlavní úlohu v dalším vývoji sehráli Brian Behlendorf a Cliff Skolnick, kteří založili e-mailovou konferenci a začali sběr úprav a jejich distribuci koordinovat.[12]

První veřejná verze s označením 0.6.2 byla vydána v dubnu 1995. Následovalo kompletní přepsání kódu (Apache2 už neobsahuje nic z původního NCSA HTTPd) a založení Apache Group, která je dnes základem vývojářského týmu. Od dubna 1996 byl Apache nejpopulárnějším serverem na internetu. V květnu 1999 běžel na 57 % všech serverů a v listopadu 2005 jeho používanost dosáhla 69 % (výsledky měření Netcraft).[12]

## **4. Analýza problému a současné situace**

### **4.1.      *Stručná historie firmy, výhled do budoucnosti***

Firma Jaroslav Schneider a spol. – Pekárna Tanvald působí na tuzemském trhu již 16let, což charakterizuje dlouhodobě stabilní vývoj firmy, která pružně uspokojuje zákazníky na základě jejich okamžitých požadavků.

Budova pekárny v Tanvaldu sloužila původně jako městské lázně, po r. 1945 jako sklad bavlny firmy SEBA Tanvald n.p. V roce 1962 byla provedena rekonstrukce a vznikla zde na svou dobu relativně moderní pekárna se dvěma průběžnými pecemi. V pekárně se vyráběl chléb a běžné pečivo.

V roce 1991 došlo k privatizaci pekárny, 1.12.1991 získal Pekárnu do ekonomického pronájmu Jaroslav Schneider. V této době byla založena firma Jaroslav Schneider a spol.- Pekárna Tanvald. V roce 1992 při tzv. velké privatizaci, byla pekárna odkoupena od FNM.

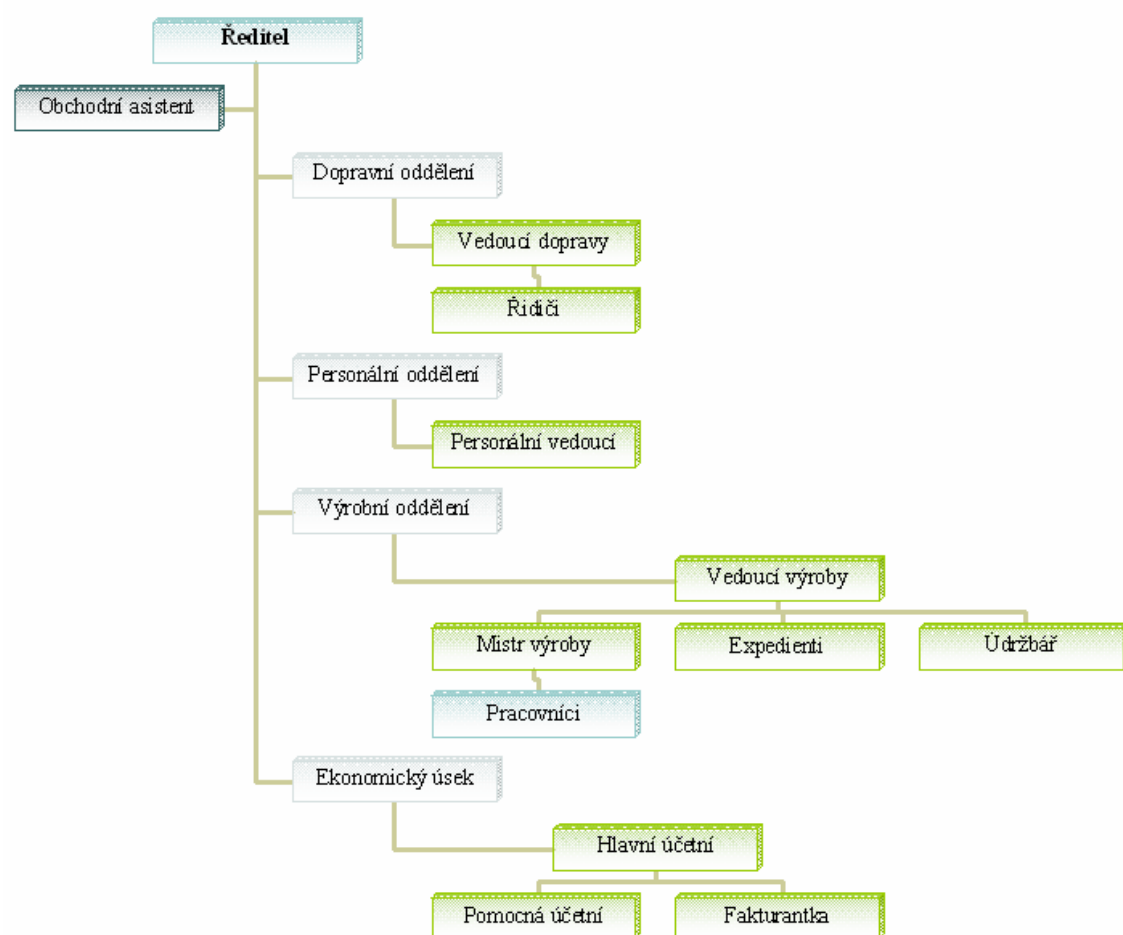
Pekárna Tanvald vyrábí v současné době cca 10 druhů chleba a koblih. Další sortiment pekařského zboží reexpeduje z jiných pekáren (Jizerské pekárny s.r.o., Pekařství Kremer s.r.o.). Pekařské výrobky jsou dodávány do obchodní sítě okresů Jablonec nad Nisou, Liberec, Semily, Mladá Boleslav a v neposlední řadě jsou zásobovány některé markety v Praze. Pekárna Tanvald denně zásobuje cca 250 odběrných míst.

V současné době obdržela firma certifikát HACCP a byl jí udělen titulu KLASA pro dva výrobky. V příštích letech je plánováno rozšíření sortimentu tzv. racionálního pečiva a jeho prodej jak u nás, tak i v rámci EU, tzn. export do SRN - příhraniční oblasti euroregionu NISA.

## 4.2. Organizační struktura

Co se týče organizační struktury firma by se dala rozdělit na čtyři základní části. A to na personální oddělení, výrobní oddělení, úsek dopravy a ekonomickou část. Jednotliví vedoucí mají své kompetence a jsou odpovědní řediteli firmy, jenž je zároveň i vlastníkem.

Právě na podnět vedoucích pracovníků jednotlivých oddělení bylo rozhodnuto o potřebě změnit stávající způsob uchovávání dat.



Obrázek 1: Organizační struktura firmy

### **4.3. Organizační procesy**

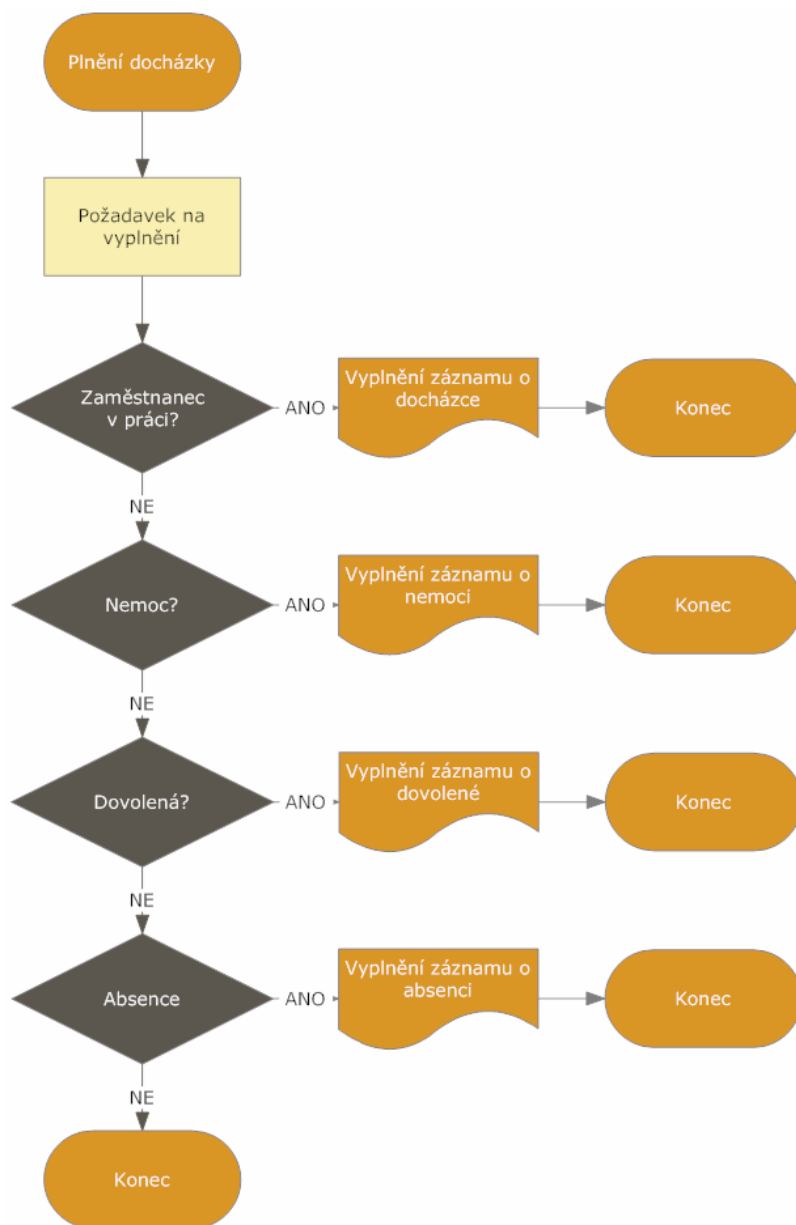
**Personální oddělení** - Toto oddělení uchovává veškeré informace o zaměstnancích, záznamy o odpracovaných hodinách, absencích, pracovních neschopnostech, upravuje pracovní pozice. Mimo agendu, týkající se zaměstnanců, má toto oddělení na starosti sklad pracovních oděvů, sklad obalů a čistících prostředků. Eviduje informace o výdeji a na jejich základě tvoří požadavek na objednání.

**Oddělení dopravy** - Náplní úseku dopravy je uchovávat informace o vozovém parku firmy. U jednotlivých vozů je potřeba evidovat řidiče, který v daný den automobil používal, počet ujetých kilometrů, přehled o tankování pohonných hmot, číslo linky kterou absolvoval, informace o příslušných garančních prohlídkách, záznamy o technických kontrolách a emisích. V případě pojistných událostí má na starosti vyřízení agendy s tím spojené a následnou evidenci případů.

**Výrobní oddělení** - Výrobní oddělení by se dalo rozdělit na další dvě části: první z nich je výroba jako taková, druhou částí je myšlena expedice. Výrobní oddělení má za úkol výrobu pečiva na základě objednávky zpracované z denního souhrnu. Na tvorbě objednávky se podílí vedoucí výroby a ředitel firmy. Náplní pracovníků expedice je nakládka pečiva jednotlivým řidičům a s tím spojený záznam o počtu naložených přepravků. Vedoucí výroby má na starosti kontrolu stavu skladu surovin a jejich následnou objednávku.

**Ekonomické oddělení** - O účetnictví firmy se stará spolu s ekonomickým oddělením externě najatý subjekt. S tímto subjektem spolupracuje především hlavní ekonom, který má vše na starosti. Ostatní zaměstnanci úseku se starají o zákazníky. Náplní jejich práce je přijetí objednávek, jejich následné zaevidování, tisk dodacích listů, vystavení faktur, dobropisu a zpracování dekády.

**Docházka** - V současné době používají všechna oddělení shodný způsob pro uchovávání informací o odpracovaném čase zaměstnanců. Tyto papírové výkazy je nutné na konci každého měsíce sčítat, což sebou nese nejen jistou časovou náročnost, ale i prostor pro vznik početního pochybení. Dále se zde naskýtá možnost vyplňování nepravdivých údajů ze strany nepoctivých zaměstnanců. Přezkoumání pravdivosti těchto údajů je zpětně těžko proveditelné.



**Obrázek 2: Diagram procesu zápis docházky**

**Kniha jízd** - Každý automobil je vybaven knihou jízd, do níž je řidič povinen zaznamenat údaje o proběhlé jízdě. Obdobně jako v případě docházky, je nutné na konci každého měsíce zkontrolovat správnost vyplnění a propočítat údaje potřebné pro další výpočet statistických údajů, jako například průměrné spotřeby jednotlivých automobilů či porovnání najetých kilometrů. Nastává zde opět problém s časovou náročností při kontrole a výpočtech.



Obrázek 3: Diagram procesu zápis do knihy jízd

#### 4.4. HW profil firmy

Vzhledem k velikosti firmy a oboru podnikání je zřejmé, že její technické vybavení v oblasti informačních technologií nebude nijak převratné. Firma nedisponuje žádnou vnitřní propojenou sítí. Otázka budování vnitřní sítě by se podle plánu měla řešit v horizontu několika následujících let a k realizaci propojení stávajícího hardwaru do jednotného celku pravděpodobně skutečně dojde.

Firma vlastní dohromady čtyři pracovní stanice, jenž jsou umístěny na jednotlivých pracovištích. Pouze výrobní oddělení z důvodu minimální práce na počítači sdílí pracovní stanici a další příslušenství spolu s oddělením personálním. To by tak mělo zůstat i nadále. Největším množstvím hardwaru disponuje ekonomické oddělení, jenž je rozděleno do dvou pracovišť. Na každém z nich je situovaná jedna pracovní stanice s příslušenstvím v podobě tiskárny pro tisk dodacích listů, faktur a ostatní nutné dokumentace. Všechny stanice, kromě přenosného počítače, který využívá dopravní oddělení, jsou připojeny do sítě internetu.

Co se týče hardwarového vybavení počítačů, je na přijatelné úrovni vzhledem k účelu, ke kterému slouží. Pokud bychom však chtěli porovnat vybavení s nynějšími standardy, zjistíme, že je průměrné a u některých stanic možná i podprůměrné. Je zřejmé, že v budoucnu při výstavbě zmiňované vnitřní sítě, bude nutné obnovit některé hardwarové prostředky.

	CPU	RAM	LAN	Operační systém
<b>Personální oddělení</b>				
<b>Výrobní oddělení</b>				
desktop 1	AMD Sepron 1.9GHz	1GB	100Mbit	Windows XP Home
<b>Dopravní oddělení</b>				
Notebook	P Celeron - 1.4GHz	512MB	100Mbit	Windows XP Home
<b>Ekonomické oddělení</b>				
desktop 2	P III - 450 MHz	258 MB	100Mbit	Windows 98
desktop 3	P Celeron - 1.6Ghz	512 MB	100Mbit	Windows 98

Obrázek 4: Hardwarový profil firmy

#### 4.5. SW profil firmy

Skladba programů na pracovních stanicích se skládá především z produktů softwarového giganta, společnosti Microsoft Corporation. Jedná se o její operační systémy Windows 98, Windows XP a aplikace MS Office.

Ekonomické oddělení má k dispozici program s databází odběratelů, jenž slouží pro tisk faktur, dodacích listů, dekádních přehledů a objednávkových sestav. Tento software je dodán na přání od firmy KoStra soft. Program mimo databáze odběratelů obsahuje databázi všech expedovaných výrobků, přehled realizovaných objednávek a archiv vystavených faktur.



Obrázek 5: Ukázka programu FAKTURACE

Stávající aplikace je v tuto chvíli pro potřeby firmy dostačující. Případná modernizace či pořízení nového systému podle firemní strategie a plánů nepřichází v nejbližší době v úvahu. Pravděpodobně na řadu přijde v průběhu dalších let a měla by navazovat na modernizaci hardwarového vybavení a s ním spojenou realizaci vnitřní sítě.

Přesto, že je systém dostačující, k jeho funkčnosti je nutno dodat pár výtek. Jednou z nich je absence možnosti tvorby objednávek prostřednictvím sítě internetu. I když



v odvětví, ve kterém se firma angažuje, na trhu, na kterém se pohybuje, a v síti maloobchodních prodejen, na něž je firma vázána, není tato služba nijak hojně využívána. Je tedy jen otázkou do budoucna, kdy si ji trh vyžádá.

Další nevýhodou tohoto programu je nemožnost nahlédnout do struktury databáze. Editace stávajících tabulek by byla jistě nemalým přínosem.

Mezi další z aplikací, vyhrazených výhradně pro ekonomické oddělení, patří program Účto2008.



Obrázek 6: Ukázka programu Účto 2008

Tento program tvoří již několik let nezbytnou součást programového vybavení firmy, což vypovídá o jeho kvalitě služeb dostačujících pro potřeby v účetnictví.

Nedostatky v dnešní době skýtá především uchovávání dat v personálním oddělení a oddělení dopravy. Obě oddělení pro svou práci používají převážně aplikace MS Office. V tomto balíku vytvořené záznamy a formuláře jsou určeny k tisku, posléze vyplněny a zaevidovány.

Oddělení doposud používají metodu ukládání dat především v papírové podobě, což sebou nese jisté nevýhody. V první řadě vzniká problém s nutnou archivací těchto spisů,

neboť je potřeba vyhradit za tímto účelem místo a pro každé nahlédnutí je třeba se fyzicky dostavit. Další problém nastává při dohledávání určitých fakt, kdy je díky nepřehlednosti a velkému množství dokumentů hledání často časově náročné a v některých případech je nutná komunikace s příslušnou osobou, jenž dokument zaevidovala.

#### **4.6.      *Shrnutí analýzy***

Hardwarové vybavení je v současném stavu na přijatelné úrovni. Do budoucna bude nutné počítat s jeho modernizací.

Co se týče softwarového vybavení, je situace poněkud horší. Pouze ekonomické oddělení využívá softwaru pro fakturaci s databází odběratelů a program Účto pro vedení účetnictví. Ostatní oddělení využívají předtištěné formuláře z textových editorů. Problém má především personální oddělení se záznamem odpracovaných hodin. Dopravní oddělení také doposud používá ukládání dat v papírové podobě. Vypracování přehledů a potřebné agendy, jenž jsou spojeny s užíváním automobilů, je časově náročné a neefektivní.

## **5. Vlastní návrhy řešení**

### **5.1.      *Databázové požadavky na systém***

Nová databáze podniku bude obsahovat informace o všech zaměstnancích firmy, kteří jsou zde zaměstnáni, o dopravních prostředcích, které se používají na rozvoz, o všech odběratelích a dodavatelích, s nimiž má firma uzavřené smlouvy, o zboží, které je zde vyráběno a o množství zásob. Tyto informace může firma v budoucnu použít i za jiným účelem jako je například tvorba statistik či jiných potřebných přehledů, péče o zákazníky atd...

Do databáze bude aktivně přistupovat pouze jeden uživatel v danou chvíli. Z tohoto důvodu nebude nutné pro danou databázi používat transakce či jiné mechanismy pro udržení konzistentního stavu databáze.

### **5.2.      *Základní požadavky na aplikaci***

Aplikace, která bude pracovat s navrhnutou databází, bude využívat především entity zaměstnanec, automobil a entity na ně navazující. Pomocí webové aplikace, kterou se k nim bude přistupovat, bude možné evidovat a uchovávat potřebná data o jednotlivých zaměstnancích a automobilech. U automobilů jde především o přehled tankování a proběhlých servisních prohlídek, tedy cosi na způsob knihy jízd. U evidence zaměstnanců se bude jednat především o jejich základní informace a o přehled odpracovaných hodin, absencí a počet vybraných dní dovolené v jednotlivých měsících.

Aplikace bude přehledná a jednoduchá. Zprvu bude sloužit pouze pro dopravní oddělení. V případě osvědčení a zaběhnutí systému se budou spolu s hardwarovou modernizací pozvolna zavádět nové funkce, které budou využívat zbytek navrhnuté databáze pro další vstupní a výstupní procesy.

### **5.3.      *Základní popis entit***

V této části bych chtěl nastínit, co budou reprezentovat jednotlivé entity, jenž bude budoucí databáze obsahovat. Dále pak ukázat fakta, na jejichž základě bude docházet k jejich vytvoření, modifikaci či zrušení. Spolu s dekompozicí databáze do třetí normální formy je zřejmé, že pravděpodobně přibudou další entity, které budou intuitivní a nebude zapotřebí jejich základního popisu. Dále zde nebudu popisovat entity spojené především s entitou Automobil. Těmi se budu zabírat v další části při řešení aplikačního rozhraní.

#### **Automobil:**

Dopravní prostředek potřebný k výkonu předmětu podnikání. Je nutné uchovávat v systému informace o jeho existenci.

Vytvoření instance: zaevidování na základě pořízení či odkupu automobilu

Modifikace: změna SPZ

Zrušení: v případě likvidace vozidla či jeho prodeje jinému fyzickému nebo právnickému subjektu

#### **Odběratel:**

Osoba, která na základě dohody platí za dodané zboží.

Vytvoření instance: zaevidování nového zákazníka

Modifikace: změna údajů (kontakty, bankovní účty, atd...)

Zrušení: v případě ukončení spolupráce (problémy s placením, požadavek ze strany odběratele)

#### **Dodavatel:**

Fyzická či právnická osoba, která na základě dohody dodává suroviny potřebné pro výrobu.

Vytvoření instance: zaevidování nového dodavatele

Modifikace: změna údajů či předmětu dodávky

Zrušení: v případě ukončení spolupráce (rozhodnutí odběratele, dodavatele, problém se zbožím či dodávkami)

**Linka:**

Trasa určená pro rozvoz produktů.

Vytvoření instance: při vytvoření nové linky

Modifikace: při změně odběratelů či při vytvoření nebo zrušení nové linky

Zrušení: v případě malé vytíženosti přerozdělení zásobovacích míst na jiné linky

**Zaměstnanec:**

Osoba najatá za účelem plnění dané pracovní náplně.

Vytvoření instance: na základě pracovní smlouvy či dohody

Modifikace: změna údajů (rodinný stav, pracovní schopnost, kontakt)

Zrušení: na základě zrušení smlouvy (dohoda, ukončení poměru, neplnění požadavků)

**Zboží:**

Produkt, jenž je předmětem výroby.

Vytvoření instance: výroba nového produktu

Modifikace: změna váhy, složení, ceny

Zrušení: ukončení výroby (poptávka)

**Zásoby:**

Suroviny a materiál potřebný pro výrobu.

Vytvoření instance: odběr nového zboží

Modifikace: cena

Zrušení: ukončení odběru (nedostupnost, při ukončení výroby)

**Docházka:**

Počet odpracovaných hodiny.

Vytvoření instance: příchodem a odchodem zaměstnance z práce

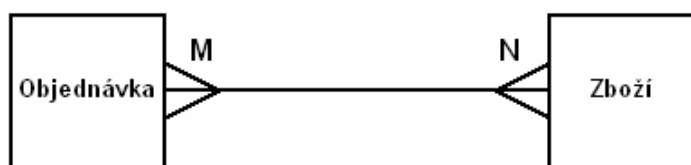
Modifikace: změna pracovní doby

Zrušení: ukončení pracovního poměru

## 5.4. Normalizace

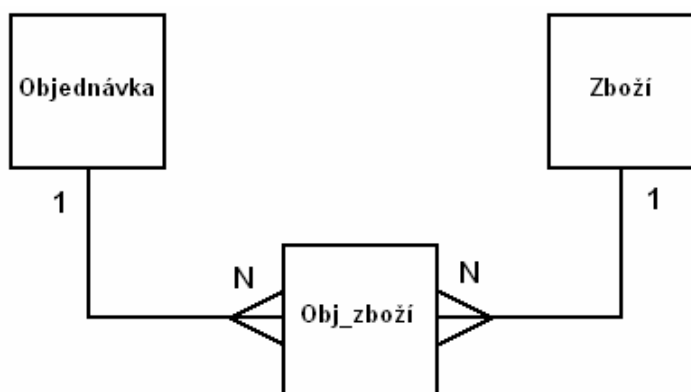
Při normalizaci datové struktury, tak aby odpovídala zvolené normalizační formě, bylo nutné u některých tabulek se vztahem N:M provést dekompozici za účelem odstranění funkční a tranzitivní závislosti. Pomocí entito-relačního modelu bych chtěl na následujících obrázcích demonstrovat odstranění těchto závislostí za účelem efektivnějšího ukládání dat a minimalizaci redundance při zachování integrity a konzistence dat.

Jedním z příkladů pro dekompozici je vztah mezi entitami Objednávka a Zboží. Vzhledem k tomu, že na jednu objednávku může být více druhů zboží a jedno zboží může být zapsáno na více objednávkách, mají tyto entity vztah M:N.



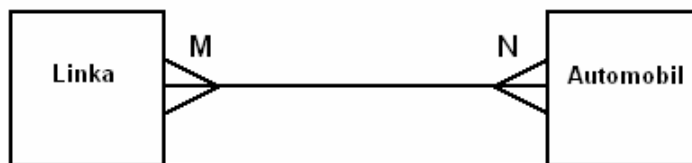
Obrázek 7: Vztah mezi objednávkou a zbožím

Pro odstranění této závislosti je potřeba navrhnout další tabulku. Nově navrženou tabulku jsem pojmenoval `obj_zboží`. Ta prezentuje objednané zboží a je vázána jak na entitu 'Objednávka', tak na 'Zboží'. S oběma entitami má vztah 1:N.



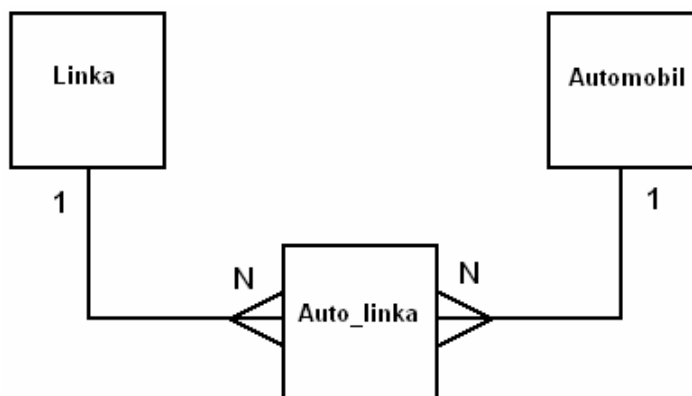
Obrázek 8: Dekompozice vztahu M:N

Dalším příkladem pro dekompozici jsou entity pojmenované Linka a Automobil. Linkou je myšlena trasa, na které je realizován rozvoz výrobků. Na jedné lince může jezdit jakékoliv auto a jeden automobil může být vyslán na kteroukoliv linku. Opět nastává situace, kdy vztah těchto tabulek je N:M.



**Obrázek 9: Vazba mezi linkou a automobilem**

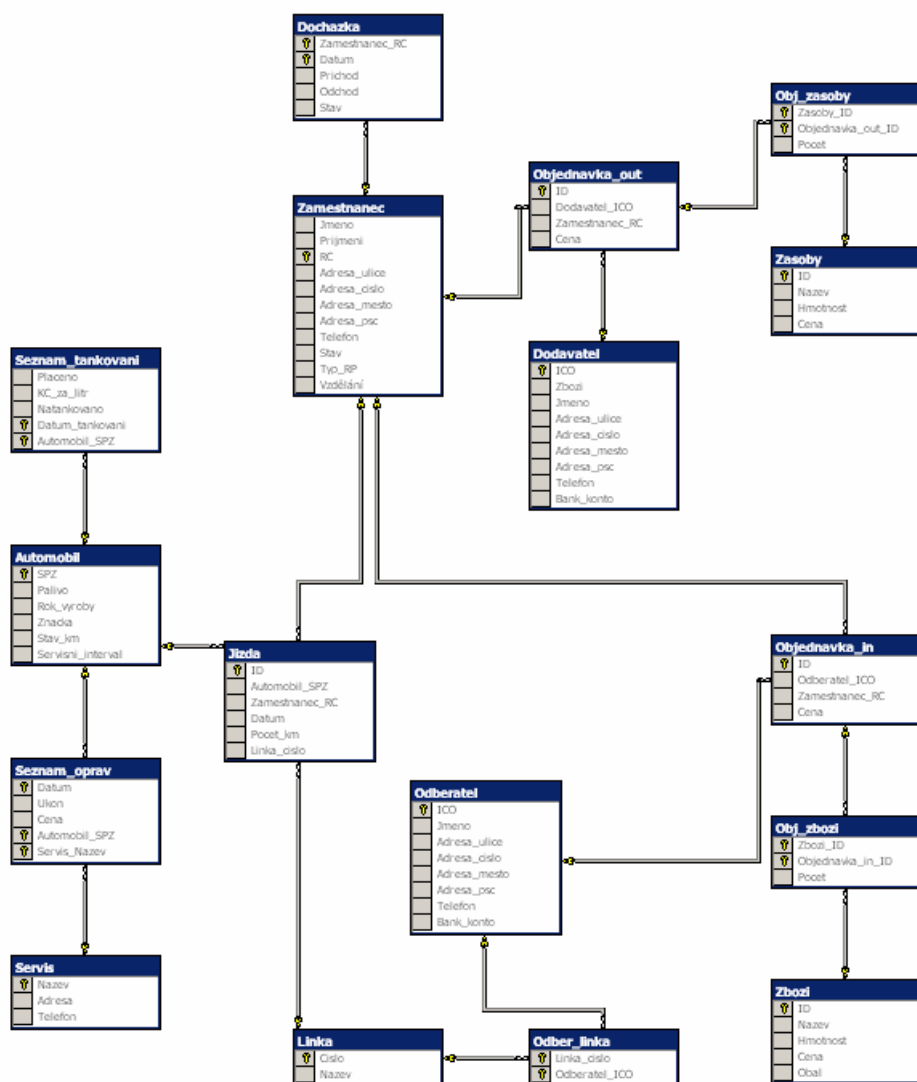
Po navržení další entity pojmenované Auto\_linka, která je vázána opět na obě stávající tabulky a s každou z nich má vztah 1:N, jsme docílili eliminace vazby M:N.



**Obrázek 10: Dekompozice vazby M:N**

## 5.5. Entitě relační diagram

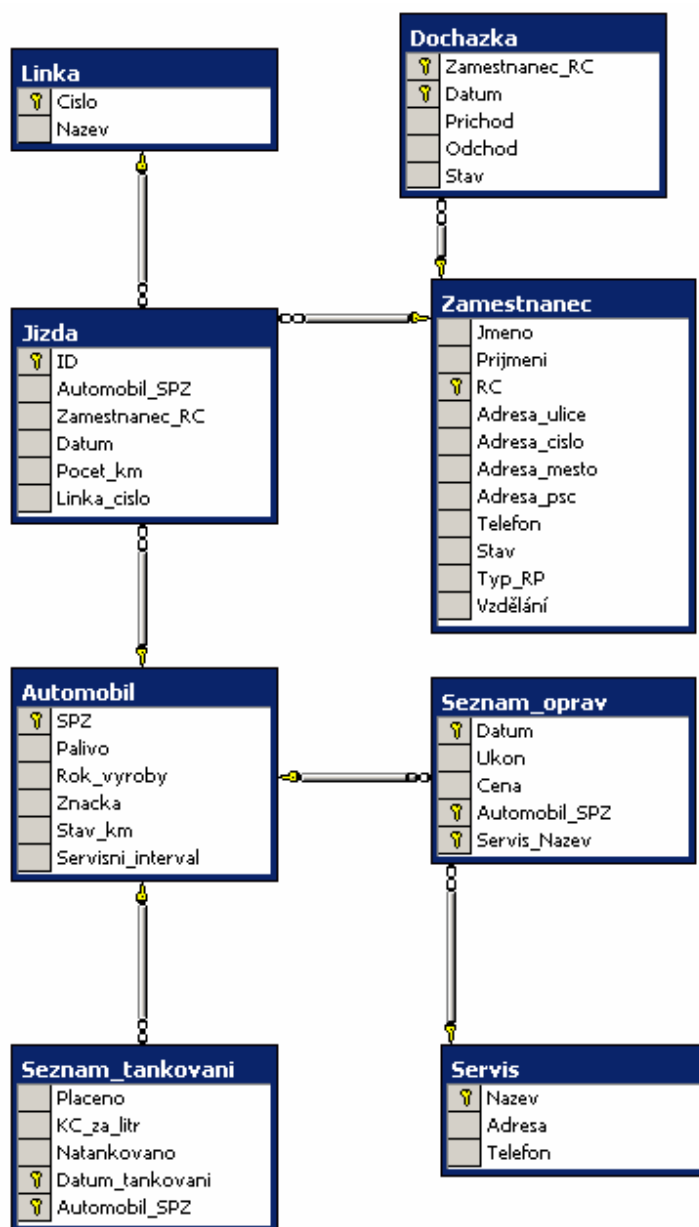
Na níže zobrazeném diagramu je znázorněna struktura dat navrhnuté databáze. Komponenty datového modelu jsou entity, jejich atributy a vztahy mezi nimi.



Obrázek 11: ER-Diagram navrhnuté databáze



Nová databáze je navrhnutá tak, aby se z ní dalo do budoucna vycházet při tvorbě dalších aplikací, či při rozšíření navrhnuté aplikace pro správu a editaci zaměstnanců a dopravních prostředků firmy. Na dalším diagramu je znázorněna provázanost entit, které budou používány již zmiňovanou navrhovanou aplikací.



Obrázek 12: ER-Diagram aplikace

## 5.6. Definice atributů entit

V následujících tabulkách nalezneme detailní popis jednotlivých entit, jenž jsou v navrhnuté databázi použity, spolu s popisem datových typů příslušných atributů a významem, jenž atribut ve skutečnosti prezentuje. V tabulkách jsou dále označeny primární a cizí klíče, které jsou potřebné pro definici vazeb mezi jednotlivými entitami. Zkratka PK (primary key) zde reprezentuje primární klíč a k označení cizího klíče je zvolen popis FK (foreign key).

TABULKA Zaměstnanec				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Jmeno	VARCHAR(10)	Not Null	PK	jméno
Prijmeni	VARCHAR(20)			příjmení
RC	VARCHAR(10)			rodné číslo
Adresa_ulice	VARCHAR(20)			ulice
Adresa_cislo	INTEGER	Null		číslo popisné
Adresa_mesto	VARCHAR(30)			město
Adresa_psc	NUMERIC(5,0)			PSC
Telefon	NUMERIC(30, 0)			
Stav	VARCHAR(10)			rodinný stav
Typ_RP	VARCHAR(3)			typ řidičského průkazu
Vzdělání	VARCHAR(10)			ukončené vzdělání

Tabulka Automobil				
Název sloupce	Datový typ	Příznak	Klíče	Význam
SPZ	VARCHAR(10)	Not Null	PK	státní poznávací značka
Palivo	VARCHAR(7)	Not Null		druh paliva
Rok_vyroby	NUMERIC(4, 0)	Not Null		rok výroby
Znacka	VARCHAR(12)	Not Null		značka výrobce
Stav_km	INTEGER	Not Null		stav kilometrů
Servisni_interval	INTEGER	Not Null		interval v km

TABULKA Linka				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Cislo	INTEGER	Not Null	PK	
Nazev	VRCHAR(50)	Not Null		

TABULKA Odběratel				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ICO	INTEGER	Not Null	PK	jméno nebo název ulice číslo popisné město PSČ kontakt bankovní spojení
Jmeno	VARCHAR(30)			
Adresa_ulice	VARCHAR(20)			
Adresa_cislo	INTEGER			
Adresa_mesto	VARCHAR(30)			
Adresa_psc	NUMERIC(5,0)			
Telefon	NUMERIC(30, 0)	Null		
Bank_konto	VARCHAR(255)	Null		

TABULKA Odber_linka				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Linka_cislo	INTEGER	Not Null	PK FK	
Odberatele_ICO	INTEGER	Not Null		

TABULKA Servis				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Nazev	VARCHAR(12)	Not Null	PK	název servisu adresa kontakt
Adresa	VARCHAR(255)	Not Null		
Telefon	NUMERIC(30, 0)	Null		

TABULKA Seznam_oprav				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Datum	DATETIME	Not Null	PK	datum opravy popis opravy cena opravy
Ukon	VARCHAR(255)	Not Null		
Cena	INTEGER	Not Null		
Automobil_SPZ	VARCHAR(10)	Not Null	PK FK	
Servis_Nazev	VARCHAR(12)	Not Null		

TABULKA Seznam_tankování				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Placeno	VARCHAR(3)	Not Null	PK FK	placeno ano/ne cena za litr počet litrů datum
KC_za_litr	INTEGER	Not Null		
Natankovano	NUMERIC(6, 2)	Null		
Datum_tankovani	DATETIME	Not Null		
Automobil_SPZ	VARCHAR(10)	Not Null		

TABULKA Dodavatel				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ICO	INTEGER	Not Null	PK	popis dodávaného zboží jméno nebo název ulice číslo popisné město PSČ kontakt bankovní spojení
Zbozi	VARCHAR(255)	Not Null		
Jmeno	VARCHAR(30)			
Adresa_ulice	VARCHAR(20)			
Adresa_cislo	INTEGER			
Adresa_mesto	VARCHAR(30)			
Adresa_psc	NUMERIC(5,0)			
Telefon	NUMERIC(30, 0)	Null		
Bank_konto	VARCHAR(30)	Null		

TABULKA Zásoby				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ID	INTEGER	Not Null	PK	identifikační číslo
Název	VARCHAR(255)	Not Null		
Hmotnost	NUMERIC(5, 0)	Not Null		
Cena	NUMERIC(6, 2)	Null		celková cena objednávky

TABULKA Zboží				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ID	INTEGER	Not Null	PK	identifikační číslo výrobku
Název	VARCHAR(30)	Not Null		název výrobku
Hmotnost	NUMERIC(5, 0)	Not Null		hmotnost v gramech
Cena	NUMERIC(4, 2)	Not Null		cena za ks
Obal	VARCHAR(1)	Not Null		baleno A/N

TABULKA Objednávka_in				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ID	INTEGER	Not Null	PK	cena objednávky
Odberatel_ICO	INTEGER	Not Null	FK	
Zamestnanec_RC	VARCHAR(10)			
Cena	NUMERIC(6,2)			

TABULKA Objednávka_out				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ID	INTEGER	Not Null	PK	cena objednávky
Dodavatel_ICO	INTEGER	Not Null	FK	
Zamestnanec_RC	VARCHAR(10)			
Cena	NUMERIC(6,2)			

TABULKA Obj_zboží				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Zbozi_ID	INTEGER	Not Null	PK FK	počet objednaného zboží
Objednavka_in_ID	INTEGER	Not Null		
Pocet	NUMERIC(4, 0)	Not Null		

TABULKA Obj_zásoby				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Zasoby_ID	INTEGER	Not Null	PK FK	počet objednaných zásob
Objednavka_out_ID	INTEGER	Not Null		
Pocet	NUMERIC(4, 0)	Not Null		

TABULKA Docházka				
Název sloupce	Datový typ	Příznak	Klíče	Význam
Zamestnanec_RC	VARCHAR(10)	Not Null	PK FK	čas příchodu čas odchodu nemoc/dovolená/absence
Datum	DATETIME	Not Null		
Prichod	DATETIME	Not Null		
Odchod	DATETIME	Not Null		
Stav	VARCHAR(8)	Not Null		

TABULKA Jízda				
Název sloupce	Datový typ	Příznak	Klíče	Význam
ID	INTEGER	Not Null	PK	identifikační číslo
Automobil_SPZ	VARCHAR(10)	Not Null	FK	
Zamestnanec_RC	VARCHAR(10)	Not Null		
Datum	DATETIME	Not Null		datum jízdy
Pocet_km	NUMERIC(3)	Not Null		počet najetých km
Linka_cislo	INTEGER	Not Null		číslo linky

## 5.7. Vytvoření tabulek

Následující kód jsem použil pro vytvoření výše definovaných entit v aplikaci MySQL. Kód definuje jednotlivé tabulky, jejich primární klíče a pomocí klíčů cizích vazby mezi nimi.

```
CREATE TABLE Zamestnanec(  
    Jmeno VARCHAR(10),  
    Prijmeni VARCHAR(20),  
    RC VARCHAR(10) Not Null,  
    Adresa_ulice VARCHAR(20),  
    Adresa_cislo INTEGER,  
    Adresa_mesto VARCHAR(30),  
    Adresa_psc NUMERIC(5,0),  
    Telefon NUMERIC(30, 0) Null,  
    Stav VARCHAR(10),  
    Typ_RP VARCHAR(3),  
    Vzdělání VARCHAR(10),  
    CONSTRAINT PK_Zamestnanec PRIMARY KEY (RC));
```

```
CREATE TABLE Automobil(  
    SPZ VARCHAR(10) Not Null,  
    Palivo VARCHAR(7) Not Null,  
    Rok_vyroby NUMERIC(4, 0) Not Null,  
    Znacka VARCHAR(12) Not Null,  
    Stav_km INTEGER Not Null,  
    Servisni_interval INTEGER Not Null,  
    CONSTRAINT PK_Automobil PRIMARY KEY (SPZ));
```

```
CREATE TABLE Linka(
    Cislo INTEGER Not Null,
    Nazev VARVARCHAR(50) Not Null,
    CONSTRAINT PK_Linka PRIMARY KEY (Cislo));
```

```
CREATE TABLE Odberatel(
    ICO INTEGER Not Null,
    Jmeno VARCHAR(30),
    Adresa_ulice VARCHAR(20),
    Adresa_cislo INTEGER,
    Adresa_mesto VARCHAR(30),
    Adresa_psc NUMERIC(5,0),
    Telefon NUMERIC(30, 0) Null,
    Bank_konto VARVARCHAR(255) Null,
    CONSTRAINT PK_Odberatel PRIMARY KEY (ICO));
```

```
CREATE TABLE Odber_linka(
    Linka_cislo INTEGER Not Null,
    Odberatel_ICO INTEGER Not Null,
    CONSTRAINT PK_Odber_linka PRIMARY KEY (Linka_cislo, Odberatel_ICO),
    FOREIGN KEY(Linka_cislo) REFERENCES Linka(cislo),
    FOREIGN KEY(Odberatel_ICO) REFERENCES Odberatel(ICO));
```

```
CREATE TABLE Servis (
    Nazev VARCHAR(12) Not Null,
    Adresa VARVARCHAR(255) Not Null,
    Telefon NUMERIC(30, 0) Null,
    CONSTRAINT PK_Servis PRIMARY KEY (Nazev));
```

```

CREATE TABLE Seznam_oprav (
    Datum DATETIME Not Null,
    Ukon VARVARCHAR(255) Not Null,
    Cena INTEGER Not Null,
    Automobil_SPZ VARCHAR(10) Not Null,
    Servis_Nazev VARCHAR(12) Not Null,
    CONSTRAINT PK_Seznam_oprav PRIMARY KEY (Datum, Automobil_SPZ,
Servis_Nazev),
    FOREIGN KEY(Automobil_SPZ) REFERENCES Automobil(SPZ),
    FOREIGN KEY(Servis_Nazev) REFERENCES Servis(Nazev));

```

```

CREATE TABLE Seznam_tankovani (
    Placeno INTEGER Not Null,
    KC_za_litr INTEGER Not Null,
    Natankovano NUMERIC(6, 2) Null,
    Datum_tankovani DATETIME Not Null,
    Automobil_SPZ VARCHAR(10) Not Null,
    CONSTRAINT PK_Seznam_tankovani PRIMARY KEY (Datum_tankovani,
Automobil_SPZ),
    FOREIGN KEY(Automobil_SPZ) REFERENCES Automobil(SPZ));

```

```

CREATE TABLE Dodavatel(
    ICO INTEGER Not Null,
    Zbozi VARVARCHAR(255) Not Null,
    Jmeno VARCHAR(30),
    Adresa_ulice VARCHAR(20),
    Adresa_cislo INTEGER,
    Adresa_mesto VARCHAR(30),
    Adresa_psc NUMERIC(5,0),
    Telefon NUMERIC(30, 0) Null,
    Bank_konto VARVARCHAR(255) Null,
    CONSTRAINT PK_Dodavate PRIMARY KEY (ICO));

```



```
CREATE TABLE Zasoby(  
    ID INTEGER Not Null,  
    Nazev VARVARCHAR(255) Not Null,  
    Hmotnost NUMERIC(5, 0) Not Null,  
    Cena NUMERIC(6, 2) Null,  
    CONSTRAINT PK_Zasoby PRIMARY KEY (ID));
```

```
CREATE TABLE Zbozi(  
    ID INTEGER Not Null,  
    Nazev VARCHAR(30) Not Null,  
    Hmotnost NUMERIC(5, 0) Not Null,  
    Cena NUMERIC(4, 2) Not Null,  
    Obal VARCHAR(1) Not Null,  
    CONSTRAINT PK_Zbozi PRIMARY KEY (ID));
```

```
CREATE TABLE Objednavka_in(  
    ID INTEGER Not Null,  
    Odberatel_ICO INTEGER,  
    Zamestnanec_RC VARCHAR(10) Not Null,  
    Cena NUMERIC(6,2),  
    CONSTRAINT PK_Objednavka_in PRIMARY KEY (ID),  
    FOREIGN KEY(Odberatel_ICO) REFERENCES Odberatel(ICO),  
    FOREIGN KEY(Zamestnanec_RC) REFERENCES Zamestnanec(RC));
```

```
CREATE TABLE Objednavka_out(  
    ID INTEGER Not Null,  
    Dodavatel_ICO INTEGER,  
    Zamestnanec_RC VARCHAR(10) Not Null,  
    Cena NUMERIC(6,2),  
    CONSTRAINT PK_Objednavka_out PRIMARY KEY (ID),  
    FOREIGN KEY(Dodavatel_ICO) REFERENCES Dodavatel(ICO),  
    FOREIGN KEY(Zamestnanec_RC) REFERENCES Zamestnanec(RC));
```

```

CREATE TABLE Obj_zbozi(
    Zbozi_ID INTEGER Not Null,
    Objednavka_in_ID INTEGER Not Null,
    Pocet NUMERIC(4, 0) Not Null,
    CONSTRAINT PK_Obj_zbozi PRIMARY KEY (Zbozi_ID, Objednavka_in_ID),
    FOREIGN KEY(Objednavka_in_ID) REFERENCES Objednavka_in(ID),
    FOREIGN KEY(Zbozi_ID) REFERENCES Zbozi(ID));

```

```

CREATE TABLE Obj_zasoby(
    Zasoby_ID INTEGER Not Null,
    Objednavka_out_ID INTEGER Not Null,
    Pocet NUMERIC(4, 0) Not Null,
    CONSTRAINT PK_Obj_zasoby PRIMARY KEY (Zasoby_ID,
Objednavka_out_ID),
    FOREIGN KEY(Objednavka_out_ID) REFERENCES Objednavka_out(ID),
    FOREIGN KEY(Zasoby_ID) REFERENCES Zasoby(ID));

```

```

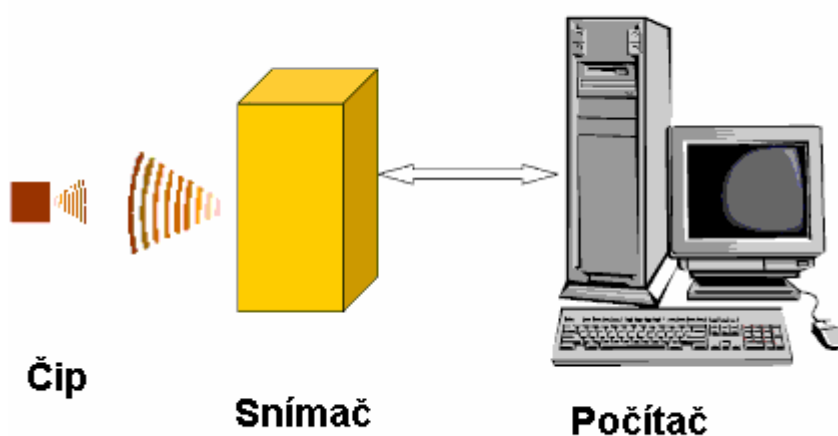
CREATE TABLE Dochazka(
    Zamestnanec_RC VARCHAR(10) Not Null,
    Datum DATETIME Not Null,
    Prichod DATETIME Not Null,
    Odchod DATETIME Not Null,
    Stav VARCHAR(8) Not Null,
    CONSTRAINT PK_Dochazka PRIMARY KEY (Zamestnanec_RC, Datum),
    FOREIGN KEY(Zamestnanec_RC) REFERENCES Zamestnanec(RC));

```

```
CREATE TABLE Jizda(  
    ID INTEGER Not Null,  
    Automobil_SPZ VARCHAR(10) Not Null,  
    Zamestnanec_RC VARCHAR(10) Not Null,  
    Datum DATETIME Not Null,  
    Pocet_km NUMERIC(3) Not Null,  
    Linka_cislo INTEGER Not Null,  
    CONSTRAINT PK_ID PRIMARY KEY (ID),  
    FOREIGN KEY(Zamestnanec_RC) REFERENCES Zamestnanec(RC),  
    FOREIGN KEY(Automobil_SPZ) REFERENCES Automobil(SPZ),  
    FOREIGN KEY(Linka_cislo) REFERENCES Linka(cislo));
```

## 5.8. *Návrh systému pro docházku*

Při řešení problému s evidencí docházky přichází v potaz využití technologie RFID. RFID je zkratka Radio Frequency Identification (radiofrekvenční identifikace). Každý zaměstnanec bude vybaven čipem (tagem) umístěným na plastové podložce, ať ve tvaru kreditní karty či přívěšku na klíče, a spojeným se spirálovou anténou, pomocí které komunikuje se snímačem. Tento snímač může pomocí softwarového vybavení filtrovat a překládat data pro informační systém.



Obrázek 13: Systém RFID

[www.hightecheid.com/images/rfidsystem.gif](http://www.hightecheid.com/images/rfidsystem.gif)

Navržená databáze bude uchovávat takto zaznamenaná data, která bude posléze informační systém zpracovávat a umožní tvořit potřebné výkazy o odpracovaném čase.

Zavedení systému bude také eliminovat nepravdivé záznamy, které se mohou ve stávajícím systému vyskytnout.

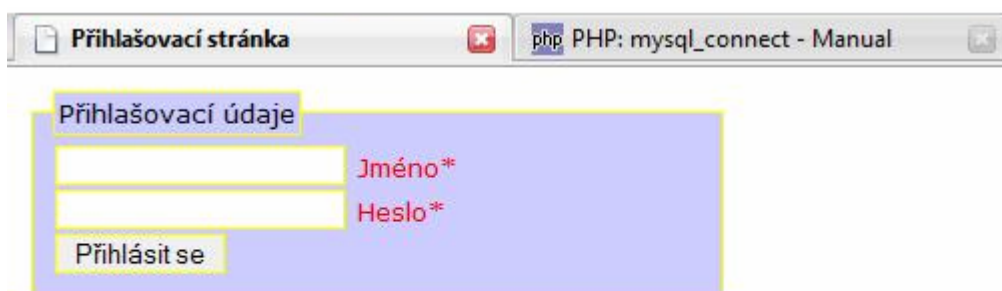
## 5.9. Řešení webové aplikace

V této kapitole se budu zabývat popisem vytvořeného webového uživatelského rozhraní. Postupně charakterizuji jednotlivé ovládací prvky, funkce portálu a přiblížím některé z příkazů MySQL a část kódu jazyka PHP použitého při tvorbě aplikačního rozhraní.

Při výběru platformy pro řešení aplikace pro přístup k navrhnuté databázi padla jasná volba na webové rozhraní a databázové prostředí MySQL. Za programovací jazyk jsem si zvolil PHP, který spolu s MySQL a Apache tvoří osvědčenou trojici volně šiřitelných technologií.

### 5.9.1. Formulář přihlášení

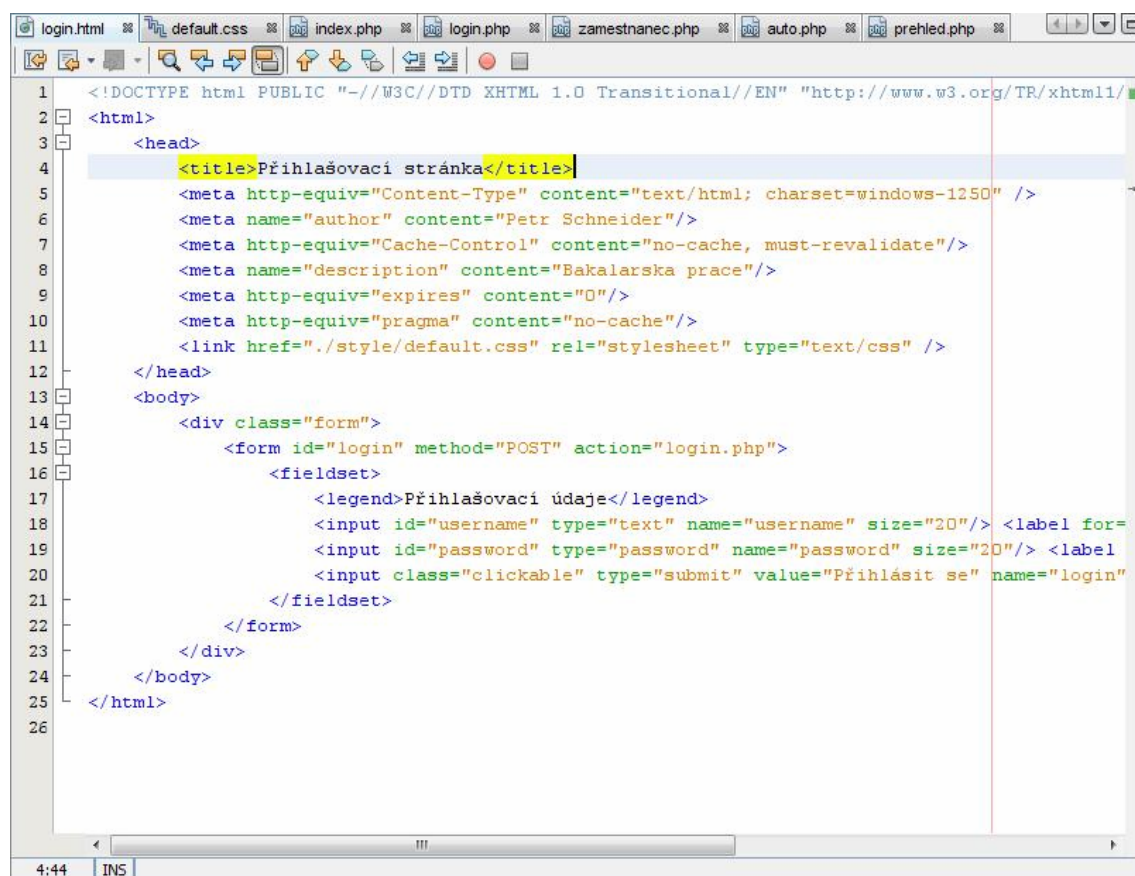
Při spuštění výchozí stránky, se uživateli zobrazí přihlašovací stránka s formulářem pro přihlašovací údaje.



The image shows a web browser window with two tabs. The active tab is titled "Přihlašovací stránka" and displays a login form. The form is enclosed in a light blue box with a yellow border. It contains the title "Přihlašovací údaje" at the top. Below the title are two input fields: the first is labeled "Jméno\*" and the second is labeled "Heslo\*". At the bottom of the form is a button labeled "Přihlásit se". The second tab in the background is titled "PHP: mysql\_connect - Manual".

Obrázek 14: Přihlašovací formulář

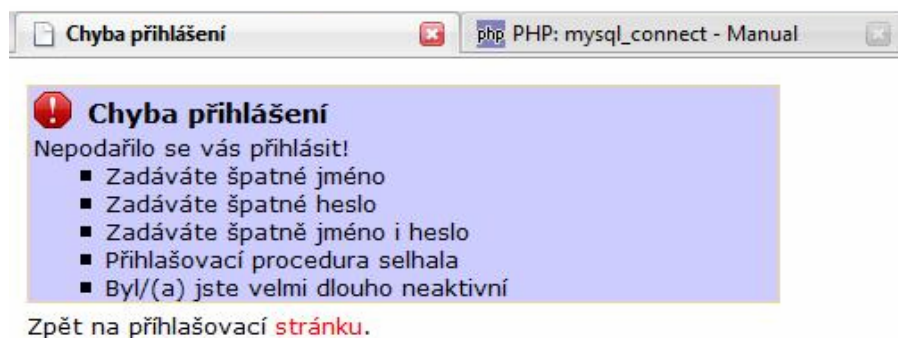
Zde je ukázka kódu výše zobrazené přihlašovací stránky login.html.



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
2 <html>
3   <head>
4     <title>Přihlašovací stránka</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
6     <meta name="author" content="Petr Schneider"/>
7     <meta http-equiv="Cache-Control" content="no-cache, must-revalidate"/>
8     <meta name="description" content="Bakalarska prace"/>
9     <meta http-equiv="expires" content="0"/>
10    <meta http-equiv="pragma" content="no-cache"/>
11    <link href="/style/default.css" rel="stylesheet" type="text/css" />
12  </head>
13  <body>
14    <div class="form">
15      <form id="login" method="POST" action="login.php">
16        <fieldset>
17          <legend>Přihlašovací údaje</legend>
18          <input id="username" type="text" name="username" size="20"/> <label for=
19          <input id="password" type="password" name="password" size="20"/> <label
20          <input class="clickable" type="submit" value="Přihlásit se" name="login"
21        </fieldset>
22      </form>
23    </div>
24  </body>
25 </html>
26
```

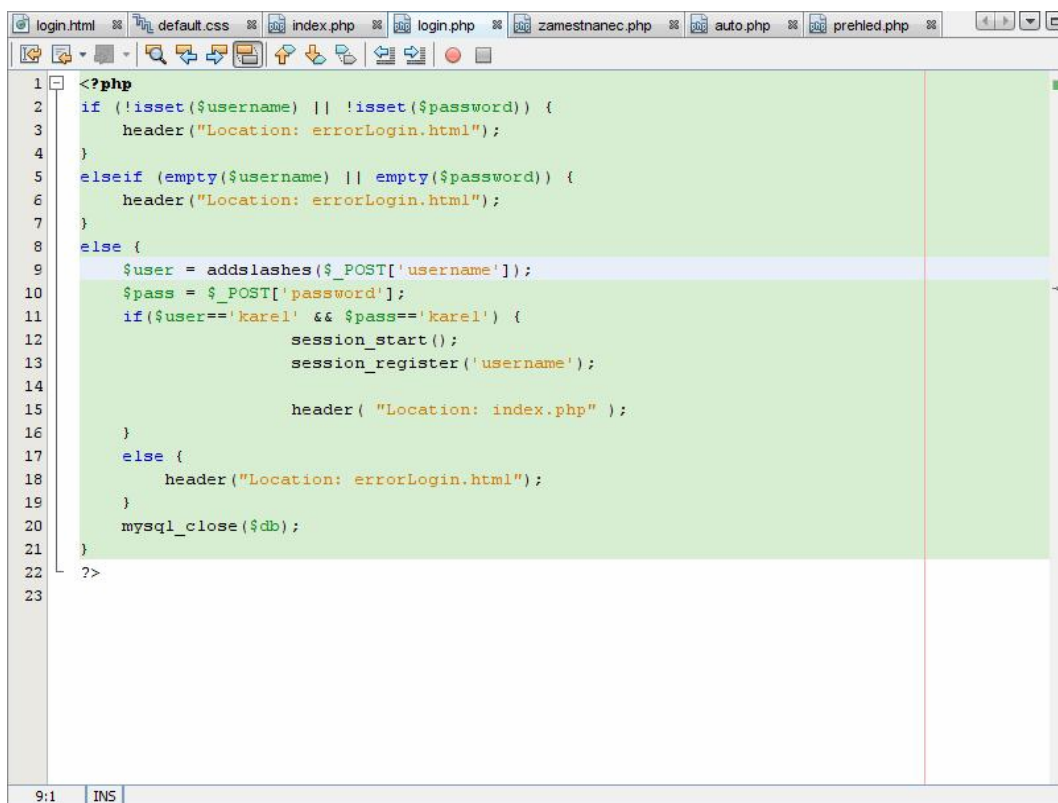
Obrázek 15: Ukázka zdrojového kódu

Vzhledem k tomu, že v danou chvíli bude k databázi přistupovat pouze jedna osoba, neobsahuje formulář registraci nového uživatele. Heslo a jméno je napevno uloženo v příslušném PHP kódu pro tuto stránku. V případě zadání špatných přihlašovacích údajů se uživateli zobrazí chybové hlášení, které umožňuje návrat na přihlašovací stránku.



Obrázek 16: Chybové přihlášení

PHP kód pro vyvolání stránky „chyba přihlášení“ při nezadání korektních přihlašovacích údajů.



Obrázek 17: Ukázka PHP kódu

### 5.9.2. Administrační část

Po zadání příslušného jména a hesla se zobrazí stránka administrace. Část administrace je rozdělena na čtyři dílčí části. Těmi jsou Zaměstnanec, Automobil, Kniha jízd a Přehled.

### 5.9.3. Zaměstnanec

Karta zaměstnance umožňuje uživateli listovat mezi zaevidovanými pracovníky, ukládat změny v jejich záznamu, vnášet do systému nové zaměstnance či smazat je v případě, že pracovník ukončí svou činnost ve firmě.

The screenshot shows a web browser window with the title 'Administrace'. The address bar shows 'MySQL :: The world's most popular op...'. The main content area has a navigation bar with links: [Zpet] [Auto] [Kniha jizd] [Přehled]. Below this is a form titled 'Zamestnanec'. The form contains several input fields with labels: Jmeno\*, Prijmeni\*, Rodne cislo\*, Ulice/c. popisne\*, Mesto/PSC\*, Telefon\*, Stav\*, Vzdelani\*, and Ridicak\*. At the bottom of the form are three buttons: Novy, Ulozit, and Smazat. There are also navigation arrows: << < > >>.

Obrázek 18: Formulář zaměstnance

Na následujícím obrázku je vidět ukázka PHP kódu pro provázanost navržené databáze s vytvořeným webovým rozhraním.



```

34 <html>
35 <head>
36 <title>administrace</title>
37 <meta http-equiv="Content-Type" content="text/html; charset=windows-1250" />
38 <meta name="author" content="Petr Schneider"/>
39 <meta http-equiv="Cache-Control" content="no-cache, must-revalidate"/>
40 <meta name="description" content="Zakalarska prace"/>
41 <meta http-equiv="expires" content="0"/>
42 <meta http-equiv="pragma" content="no-cache"/>
43 <link href="style/default.css" rel="stylesheet" type="text/css" />
44 </head>
45 <body>
46 <div style="float: right; margin: 0 20px 0 0;">
47 <a href="logout.php">[logout]</a>
48 </div>
49 <div>
50 <a href="index.php">[Zpet]</a> <a href="auto.php">[Auto]</a> <a href="prehled.php">[Prehled]</a>
51 </div>
52 <table>
53 <tr><td width="400px" valign="top">
54 <div class="form">
55 <?php
56 $query = "select count(*) as pocet from zamestnanec";
57 $result=mysql_query($query, $db);
58 $row = mysql_fetch_array($result);
59 if ($naznam > $row['pocet'] - 1) {
60 $naznam = $row['pocet'] - 1;
61 }
62 if (isset($last)) {
63 $pocet = $row['pocet'];
64 }
65 $query = "select jmeno, prijmeni, rc, adresa_ulice, adresa_cislo, adresa_mesto, adresa_psc, telefon, typ_rp, stav, vadelani from zame";
66 mysql_query("SET NAMES 'cp1250'");
67 if ($novy != true) {
68 $result=mysql_query($query, $db);
69 }
70 $rowCount = mysql_num_rows($result);
71 if ($rowCount > 0) {
72 $row = mysql_fetch_array($result);
73 $jmeno = $row['jmeno'];
74 $prijmeni = $row['prijmeni'];
75 $rc = $row['rc'];
76 $ulice = $row['adresa_ulice'];
77 $ulice_cislo = $row['adresa_cislo'];

```

Obrázek 19: Ukázka kódu PHP

#### 5.9.4. Automobil

Stránka automobil má vesměs stejné funkce a slouží ke stejnému účelu, jako předchozí formulář zaměstnance, a to tedy hlavně k evidenci. Servisní interval zde uvádí počet kilometrů, po jehož najetí by měl být automobil přistaven k plánované garanční prohlídce. Software, jenž by pracoval s navrženou databází, by na základě tohoto údaje a údajů o aktuálním stavu kilometrů jednotlivých automobilů upozorňoval na blížící se povinné kontroly.

Administrace MySQL :: The world's most popula...

[Zpet] [Zamestnanec] [Kniha jizd] [Prehled]

Automobil

SPZ\*

Palivo\*

Rok výroby\*

Znacka\*

Stav KM\*

Servisni interval

<< < > >>

Novy Ulozit Smazat

**Obrázek 20: Formulář Automobil**

Připomínky by se netýkaly pouze plánovaných garančních oprav, ale také návštěv pracovišť technické kontroly. Tyto informace by systém zjistil na základě seznamu provedených oprav jednotlivých dopravních prostředků. Pro editaci servisních prací slouží formulář servis na stránce kniha jízd.

### 5.9.5. Kniha jízd

Formulář kniha jízd umožňuje uživateli zaznamenávat informace potřebné k vedení agendy spojené s provozem automobilů. Na stránce se nacházejí dva další formuláře, již zmíněný servis a tankování. Oba se váží na aktuálně vybraný automobil v knize jízd a umožňují zadat další doplňující informace.

The screenshot shows a web browser window with the title 'Administrace'. The address bar shows 'MySQL :: The world's most popular op...' and the page is loaded from 'php PHP: Hypertext Preprocessor'. The browser tabs include 'meebo.cz - online klient pro icq, googl...'. The main content area has a navigation bar with links: [Zpet], [Zamestnanec], [Auto], [Prehled]. Below the navigation bar, there are three forms, each with a yellow border and a light blue background.

**Kniha jízdy**

JNA 06-59 Automobil\*  
 Novak Ridic\*  
 8 - Jablonec Linka\*  
☐ aktualni datum\*  
 Datum\*  
 Pocet najetych KM\*  
 Zadat

**Servis**

☐ aktualni datum\*  
 Datum\*  
 Ukon\*  
 Cena\*  
 Mercedes Servis\*  
 Zadat

**Tankovani**

☐ aktualni datum\*  
 Datum\*  
 Cena\*  
 Kc/l\*  
 Litru\*  
 Zadat

Obrázek 21: Formulář knihy jízd

## 5.9.6. Přehled

Formulář na stránce přehled nabízí uživateli možnost zobrazit u zvoleného automobilu informace týkající se například čerpání pohonných hmot.

The screenshot shows the same web browser window as before. The main content area now displays the 'Prehled' form, which is a single large form with a light blue background and a yellow border. The form has a title 'Prehled' and a navigation bar with links: [Zpet], [Zamestnanec], [Auto], [Kniha jízdy].

**Prehled**

JNA 67 02 Automobil\*  
 Leden Mesic\*  
 2008 Rok\*  
☒ servis\*  
☐ tankovani\*  

1.2.2008	STK/ETK	135835 Km	3000.-	Adam
1.3.2008	Vymena oleje	345983 Km	6000.-	Novak

 Zobrazit

Obrázek 22: Formulář přehled

## 6. Přínos návrhů řešení

Důležitým přínosem pro firmu je komplexně navrhnutá databáze, na jejímž základě se dá do budoucna stavět. Pomocí takto uložených dat může firma tvořit statistiky či jinak potřebné přehledy.

Přínos návrhu datové aplikace pro zvolenou firmu by se měl zprvu odrazit při zefektivnění probíhajících procesů především v personálním a dopravním oddělení. Díky navrhnutému řešení by se tyto procesy měly automatizovat.

U systému evidence docházky již nebude nutné denně zapisovat odpracované hodiny, což přinese ulehčení personálnímu oddělení, které nebude muset koncem každého měsíce zdoluhavě přepočítávat a kontrolovat jednotlivé výkazy o odpracovaném čase, které se doposud používají. Dojde tak i k eliminaci vyplňování nepravdivých údajů.

Přínos v podobě konce nutných výpočtů čeká i dopravní oddělení. Díky informacím o jednotlivých jízdách, které se budou pravidelně do systému zadávat, odpadnou zdoluhavé výpočty a zmizí prostor pro možné vzniklé početní chyby. Shromážděná data nám dají detailní přehled o jednotlivých automobilech. Počínaje dny, kdy bylo auto v provozu, kdo jej využíval, počet najetých kilometrů daný den, informace o provedeném čerpání pohonných hmot a servisních kontrolách. Pomocí těchto informací můžeme zjistit, kdy je potřeba automobil přistavit například k další garanční prohlídce či kontrole ETK/STK.

## 7. Závěr

Tato práce poskytla analýzu současného stavu vybraných procesů ve firmě. Na jeho základě byl vytvořen entito-relační diagram pro komplexní databázi, a dále bylo navrženo grafické uživatelské rozhraní s přístupem přes webový prohlížeč.

V rámci této práce byl vytvořen návrh datové aplikace, který má posloužit pro rozvoj společnosti zabývající se prodejem a rozvozem potravinářských výrobků. Celkový přínos tohoto projektu je kladný, zvýšil efektivnost práce zaměstnanců, pro něž byl systém navrhnout.

Navržený systém není nikterak omezen v dalším rozvoji a v průběhu jeho životnosti se může rozšiřovat a přizpůsobovat aktuálním požadavkům.

Co se týče stanovených cílů, mohu říci, že se mi jich povedlo dosáhnout.

## 8. Seznam použité literatury

### 8.1. Literatura

- [1] JAMES, Michael. *Návrh databází*. [s.l.] : [s.n.], 2006. 408 s. ISBN 80-247-0900-7.
- [2] MERUNKA, Vojtěch. *Objektový přístup v databázových systémech*. 2002. ISBN: 80-213-0882-6.
- [3] PÍSEK, Slavoj. *Databáze v Accessu*. [s.l.] : [s.n.], 2003. 88 s. ISBN 80-247-0572-9.
- [4] POKORNÝ, Jaroslav. *Databázové systémy 2*. [s.l.] : [s.n.], 2007. 190 s. ISBN 978-80-01-03797.
- [5] JAMES, Michael. *Návrh databází*. [s.l.] : [s.n.], 2006. 408 s. ISBN 80-247-0900-7.
- [6] RIESSLER, Petr. *Databáze a programování*. 2000. ISBN: 80-214-1778-1.
- [7] RIORDAN Rebecca M.. *Vytváříme relační databázové aplikace*. 2000. ISBN: 80-7226-360-9.
- [8] SIMPSON, Alan. *Access 97 : [kompletní popis všech částí databázového systému včetně možnosti zpřístupnění databáze po Internetu a tvorby jednoduchých programů ve VBA s řadou praktických příkladů a rad : pro všechny kategorie uživatelů]*. [s.l.] : [s.n.], 1998. 966 s. ISBN 80-7169-612-9.
- [9] STEVEN, Roman. *Microsoft Access : návrh a programování databází : co potřebujete opravdu vědět o tvorbě databází*. [s.l.] : [s.n.], 1990. 250 s.

- [10] VAUGHN William R.. Visual Basic pro SQL server : průvodce tvorbou databázových aplikací.1998. ISBN: 80-7226-085-5.
- [11] WOLFGANG, Kristen. *Caché : databáze postrelačního typu a tvorba aplikací*. [s.l.] : [s.n.], 2005. 400 s. ISBN 80-251-0491-5.

## **8.2. Internetové zdroje**

- [12] *Http://cs.wikipedia.org* [online]. 2002-2008 [cit. 2008-05-11]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/Apache\\_HTTP\\_Server](http://cs.wikipedia.org/wiki/Apache_HTTP_Server)>.
- [13] *Http://cz.php.net/* [online]. c2001-2008 [cit. 2008-05-10]. Dostupný z WWW: <<http://cz.php.net/>>.
- [14] *Http://dev.mysql.com/* [online]. c1995-2008 [cit. 2008-05-10]. Dostupný z WWW: <<http://dev.mysql.com/>>.
- [15] *Http://www.owebu.cz* [online]. 2001-2008 [cit. 2007-05-15]. Dostupný z WWW: <<http://www.owebu.cz/databaze/vypis.php?clanek=334>>.
- [16] *Http://www.tvorba-webu.cz* [online]. c2003-2008 [cit. 2008-05-10]. Dostupný z WWW: <<http://www.tvorba-webu.cz/php/historie.php>>.
- [17] *Interval.cz* [online]. 2002 [cit. 2008-05-12]. Dostupný z WWW: <[interval.cz/clanky/webml-datove-modelovani](http://interval.cz/clanky/webml-datove-modelovani)>.
- [18] MATUNA. *Http://www.sweb.cz/martin.matuna/* [online]. 2005 [cit. 2008-05-16]. Dostupný z WWW: <<http://www.sweb.cz/martin.matuna/>>.
- [19] *Www.dbsvet.cz* [online]. c2004 [cit. 2008-05-12]. Dostupný z WWW: <[www.dbsvet.cz/view.php?cisloclanku=2004061601](http://www.dbsvet.cz/view.php?cisloclanku=2004061601)>.

## 9. Seznam obrázků

Obrázek 1: Organizační struktura firmy .....	18
Obrázek 2: Diagram procesu zápis docházky .....	20
Obrázek 3: Diagram procesu zápis do knihy jízd .....	21
Obrázek 4: Hardwarový profil firmy .....	22
Obrázek 5: Ukázka programu FAKTURACE .....	23
Obrázek 6: Ukázka programu Účto 2008 .....	24
Obrázek 7: Vztah mezi objednávkou a zbožím .....	29
Obrázek 8: Dekompozice vztahu M:N .....	29
Obrázek 9: Vazba mezi linkou a automobilem .....	30
Obrázek 10: Dekompozice vazby M:N .....	30
Obrázek 11: ER-Diagram navrhnuté databáze .....	31
Obrázek 12: ER-Diagram aplikace .....	32
Obrázek 13: Systém RFID .....	43
Obrázek 14: Přihlašovací formulář .....	44
Obrázek 15: Ukázka zdrojového kódu .....	45
Obrázek 16: Chybové přihlášení .....	46
Obrázek 17: Ukázka PHP kódu .....	46
Obrázek 18: Formulář zaměstnance .....	47
Obrázek 19: Ukázka kódu PHP .....	48
Obrázek 20: Formulář Automobil .....	49
Obrázek 21: Formulář knihy jízd .....	50
Obrázek 22: Formulář přehled .....	50